AD-A008 457

WHY THINGS ARE SO BAD FOR THE
COMPUTER-NAIVE USER

William C. Mann

University of Southern California

Prepared for:

Office of Naval Research
Advanced Research Projects Agency

March 1975

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/RR-75-32	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER *AD-A008 457*
4. TITLE (and Subtitle) WHY THINGS ARE SO BAD FOR THE COMPUTER-NAIVE USER		5. TYPE OF REPORT & PERIOD COVERED Research
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William C. Mann		8. CONTRACT OR GRANT NUMBER(s) N00014 75 C 0710
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute 4676 Admiralty Way Marina Del Rey, CA 90291		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order #2930
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, Virginia 22209		12. REPORT DATE March 1975
		13. NUMBER OF PAGES 13
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) -------------
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Document available for public release and sale; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Submitted to the National Computer Conference - May 1975

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer user, command language, syntax, semantics, interface, computer language, computation skills, conceptual complexity, concept representation, natural language.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(OVER)

DD FORM 1 JAN 73 **1473** EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0103-014-6601

20. ABSTRACT

Many people who use computers, or have tried to use them, find them extremely difficult to master, understand, interact with. Computers have a well-earned reputation for alienness and intractability.

Much of the difficulty arises from the prevailing ways that computer programs communicate with people. Computer professionals have been preoccupied with commands and command languages, to the exclusion of the kinds of communication that people use most of the time with each other. To make use of a computer, people are forced into an unfamiliar command-oriented organization, and many cannot make this extreme transition.

By expanding the scope of human-computer interaction methods to include other styles of interaction, computer systems can be made more compatible with the computer-naive potential user. This enhanced compatability will open up new applications in which computer-naive people make direct use of computers to extend their working abilities. This paper identifies the gap between today's dominant styles of person-computer communication and interpersonal communication, and suggests the developments needed to make computers more people-compatible.
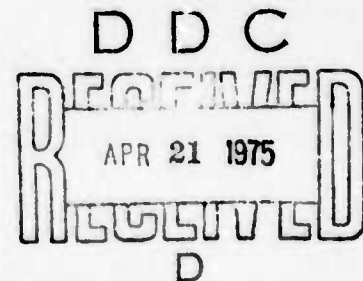
William C. Mann

Why Things Are So Bad for the Computer-Naive User

D D C

APR 21 1975

D

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

ABSTRACT

Many people who use computers, or have tried to use them, find them extremely difficult to master, understand, interact with. Computers have a well-earned reputation for alienness and intractability.

Much of the difficulty arises from the prevailing ways that computer programs communicate with people. Computer professionals have been preoccupied with commands and command languages, to the exclusion of the kinds of communication that people use most of the time with each other. To make use of a computer, people are forced into an unfamiliar command-oriented organization, and many cannot make this extreme transition.

By expanding the scope of human-computer interaction methods to include other styles of interaction, computer systems can be made more compatible with the computer-naive potential user. This enhanced compatability will open up new applications in which computer-naive people make direct use of computers to extend their working abilities. This paper identifies the gap between today's dominant styles of person-computer communication and interpersonal communication, and suggests the developments needed to make computers more people-compatible.

WHY THINGS ARE SO BAD FOR THE COMPUTER-NAIVE USER

This paper is about people's use of computers to get work done. Computers are tools that operate on symbols. Unlike many other tools, such as eggbeaters, telephones and automobiles, most people in our culture seem to regard computers as alien, mysterious and inherently difficult to use. While some of this view is based on mere hearsay, it is nonetheless held by many who have had some experience with computer systems.

Why? Because many computer systems are in fact alien, mysterious and inherently difficult to use. This is true even for interactive systems, where the opportunities for accomodation of users are the greatest and the technical history is the richest.

When people first attempt to use computers, they find that they must communicate with them. They are confronted with a variety of "interfaces," each possessing a "language" and some conventions for abbreviation, spelling help, prompting and the like. There are too many such interfaces, and they are all different in arbitrary, unimportant and hard-to-remember ways, but these are superficial problems. A couple of iterations of thoughtful system re-implementation could reconcile them.

Much worse, the interfaces share a common core of methods and demands which is itself alien, unlike any language known to natural man. This problem is not at all superficial-- it arises out of fundamental inadequacies in our usual methods for organizing computations and specifying systems. Depending on one's preference, it can be described as a problem in technology or a problem of the subculture that carries the technology. We will describe it here as a technical problem.

Command Languages

When we look at the facilities which system interfaces typically provide to their users, we find that they are almost exclusively **command languages**. Users can issue commands and supply command parameters. Other facilities of the interface are designed to aid in the selection and use of these commands. There is very little else.

The problem with command language forms is not that they are formally incomplete; they are not. Neither is it that they limit the users' access to

the full diversity of the machines, although this happens. Neither are the problems in any fundamental way problems of efficiency or cost.

The basic difficulty, the source of this alienness and intractability, is that commands are an extremely narrow, limiting subset of people's familiar range of expression. Radical specialization is required if a person is to express his desire for the accomplishment of some task entirely as a combination of commands.

There's More to Communication with People
Than Just Commands

If I wanted you to prepare an index of the books and papers in my office, I would first describe to you how I use them. This would give you some clues which would help you to distinguish a useful index from a non-useful one. I would talk about the urgency of the work, which would give you some clues about how to share your attention between this task and others. Then I would show you how to access the collection, and state whatever properties I expected the final results to have - order, format, index terms etc. At every point I would expect you to acknowledge what you satisfactorily understood, and to discuss and obtain clarification on the rest.

The result of our discussion would not be a complete task specification. Many aspects would remain unspecified. I would expect that as you performed the work, some unexpected choices would become necessary, and that you would make some of those choices yourself based on your knowledge of my goals.

Some Representative Interactive Computing Tasks
and Processors*

Task	Processors
Information Retrieval	Dialog (R)
System Manipulation	Timesharing Operating Systems (many)
Programming	APL, PL/1
Text Editing	(many)
Text Formatting	Runoff, Script
Computer Network Manipulation	Telnet, FTP
Message Processing	Sndmsg, Readmail

----

* Most of these are PDP-10 programs, some of them peculiar to the ARPANET. But the problem is much more widespread, not confined to one class of machines or one subculture of computing.
Dialog is a trademark of the Lockheed Corporation.

Goals and Purposes

"I want to be able to find things I have already read, and maintain a list of stuff I haven't read yet by topic, so that I can easily pick things to read."

Examples

"So you would index Speech Acts under Searle, and Ordinary-language Philosophy and Illocution."

Description

"The index should include several entry regions, by author and title and so forth, and a region of citations."

Clarifications

"I want all the papers indexed."
"Do you mean the papers in the journals too?"
"No."
"OK."

Hypothetical Conditions

"Suppose I find two different papers with the same title and author."

Functional Descriptions

"The location code tells where the item is physically, within a couple of feet of shelf."

Analogies and Comparisons

"The authors section is like the white pages, and the subject section is like the yellow pages."

Similarities and Differences

"It's like a library card catalog, except that we're not using code numbering or cards."

Refused Commands

"' can't index the papers this week."

Sample Statements from the Two-Person Indexing Task

I estimate that less than 5% of our communication would be in commands, requests or other directives. We would exchange descriptions of objects and processes, convey concepts using examples, and negotiate the meanings of terms. We would discuss hypothetical, perhaps even impossible situations. You might not accept all of my "commands," not even all of the sensible ones. We would talk about goals and purposes for objects, actions and steps in the process. Analogies, similarities, differences and comparisons would all be used to express ideas.

The point of this whole elaborate example is to demonstrate how easy it is to specify an information processing task by using language which hardly includes any commands or command parameters at all. It is even reasonable to expect satisfactory results (om a suitable interpreter) based on such a specification.

Consider the hypothetical examples above, from this indexing task. We can observe several things about them:

1. There is no easy translation for any of them into English commands.

2. They represent kinds of communication that are not commonly provided for at all in people's access to computer systems.

3. They are all different in purpose and apparent effect.

4. They are natural to the task, as done by people

5. Several of them represent initiatives by the receiver of the task.

The corresponding lessons for design of man-machine communication are:

1. Accomodation of command languages (even an English language command subset) is a strong restriction on people's capabilities. It brings difficulties that are not part of the basic task.

- 2. Computer systems are commonly deficient in meeting the information needs of users.

3. Using commands is a small part of people's communication repertoire. Since people cannot translate these other kinds of communication into commands easily, (1 above) , non-command interaction methods need to be designed into systems.

4. There are lots of different opportunities for major improvement.

5. It might be helpful to give the system more initiative in communication than they typically have in current practise.

It is this diverse kind of communication that is familiar and tractable to the computer-naive person. Most people who could express this task in ordinary language would be unable to reexpress it entirely as a sequence of commands.* The difficulty of translating a task into commands is of course not confined to this example or this class of problems.

For many people, being forced to translate their desires into commands is an unpleasant imposition, and is even felt as an attack on their established competence. For some, the difficulty is never overcome, and for others the use of computers becomes a thing to be minimized, whatever the supposed benefits might be.

Designing Systems for Ordinary People

As a long-range goal, we should seek to develop interaction methods that accept the full range of expression suggested above. Below we consider how to expand our stock of tools in these ways.

I - A Design Approach for Today

We need to avoid overly optimistic expectations on systems currently in development - those that are based on command-language communication. Developing command languages and command help facilities is important, but mainly for the computer professionals and heavy users who have already adapted to command language demands.

Merely making easier command languages will not help the really computer-naive users very much, since it does not really address their problems. So in the near term we need ways to supplement the command structures of users' languages. Later we should seek ways to include commands as part of much more comprehensive schemes. We can expect a continuing expansion of the facilities in systems for the computer-naive. How should that expansion be done?

----

* In fact, the computer professionals are the only group who can regularly do it on a variety of tasks. They provide, in the systems they build, tools that are congenial to their own conceptual habits, and alien to nearly everybody else's.

Three properties are especially important for the systems and programs currently being designed:

1. Language structure that admits non-commands from the user.

2. Intentional imitation of human dialogue at the users' interface. In order to end up accepting a broad diversity of the kinds of communication that people use freely, many steps of imitation will be needed. (This is not the same as shifting to natural language. Formal languages can be diversified in the right ways.) We can start now by strengthening the declarative and descriptive parts of languages, allowing alternative equivalent forms of expression and imitating the control structures seen in dialogue.

3. Continuity with the currently available best practise. Although we must move beyond commands-alone, commands will always be with us and be important. There are extreme contrasts in the command styles of today's systems, and there are too many new, badly done interfaces.

II - Design Changes in the Future

There is a general lack of the kind of detailed knowledge of human communication that we need. Many of the things that people do in communication, including the entire list indicated above, are not understood well enough to support imitation. We lack a good scientifically established model for the simplest case: successful communication between two cooperating people.

Having a good model of people's communication activities would be beneficial far beyond the sphere of computing. It would be a significant advance for psychology, for education, for the medical knowledge of communication disorders, for documentation and information dissemination, for linguistics and other disciplines as well. It would speak to the strong intellectual interest of our century in epistemology. And it might eventually guide people into more effective communication with each other.

But just on the basis of the potential benefits in computing alone, making direct use of computers feasible and comfortable for broad classes of people, research in modeling human communication processes deserves a far higher national priority than it currently has.

There are a few active research projects that are building the right kinds of models of human communication capabilities, in a framework relevant to computer system design. For example, in the automated consultant work at Stanford Research Institute, dialogue between a knowledgeable mechanic and an apprentice is being analyzed, with the intention of allowing similar dialogues to take place between an apprentice and an automated knowleage ble mechanic. (1)

In the Sophie instruction system currently being developed at Bolt, Beranek and Newman, Inc., a trainee in electronic fault diagnosis and repair interacts with a computer program. (2) The program represents and manipulates a circuit with hidden faults. The trainee can discuss hypothetical conditions, seek evaluations of guesses (of faults), cause measurements to be made, and ask questions, all in English.

In the Protocol Analysis System II, developed at Carnegie-Mellon University, a program analyzes a transcript of a student's remarks made while solving a problem, and thereby follows his progress. (3)

The Dialogue Process Modeling work at Information Sciences Institute is building computer-program models of specific two-person interactions in English, analyzing the communication effects that people have on each other.

Each of these efforts involves a computer responding to the language forms that people commonly use. None of them are restricted to command language interaction. Several of them have demonstrated capacity for effective response to a significant portion of English expression.

These projects, and others like them, can multiply our understanding of the details of communication as people do it. The hope is that in the future there will be comfortable communication, with the kind of diversity that human dialogue has, commonly available on computer systems.

References

(1)   Deutsch, Barbara G., "The Structure of Task Oriented Dialogs," Technical Note 90, SRI Project 1526, Stanford Research Institute, Menlo Park, Calif. 94025; appeared in **Proceedings IEEE Speech Symposium**, Carnegie-Mellon University, Pittsburgh, Pa., April 15-19, 1974.

(2)   Brown, John Seely, Richard R. Burton, and Alan G. Bell, SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI), Bolt, Beranek and Newman, Inc., Artificial Intelligence, Final Report (A.I. Report No. 12), 1 March 1974, pp. 24-25, 96.

(3)   Waterman, D. A., and A. Newell, PAS-II: An Interactive Task-Free Version of An Automatic Protocol Analysis System, Department of Computer Science, Carnegie-Mellon University, June, 1973.

(4)   Winograd, Terry, "A Procedural Model of Language Understanding," **Computer Models of Thought and Language**, W. H. Freeman and Company, San Francisco, California, 1973, pp. 152-186.